Fixpoints vs Moore Families

Zhang, Fuyuan; Nielson, Flemming; Nielson, Hanne Riis

Published in:
Proceedings of SOFSEM 2011

Publication date:
2012

Fixpoints vs Moore Families

Fuyuan Zhang, Flemming Nielson, Hanne Riis Nielson

DTU Informatics, Technical University of Denmark, Lyngby, Denmark
{fuzh,nielson,riis}@imm.dtu.dk

Abstract. Model checking and static analysis are both successful approaches to the analysis of IT systems and it has been shown that many static analyses can be reduced to model checking. Recent results show that CTL model checking can be reduced to static analysis and that the set of satisfying states of a CTL formula can be described as the least element in a Moore family of acceptable sets of states for the static analysis. Turning the attention to the μ-calculus we are able to generalise this result to the alternation-free fragment whereas even for the fragment of alternation depth 2 we show that the fixed point characterisation cannot be recast as a Moore family property.

1 Introduction

Both *model checking* [1, 5] and *static analysis* [7] are prominent approaches to detecting software errors. Model Checking is a successful formal method for verifying properties specified in modal logics with respect to transition systems. Static analysis is also a powerful method for validating program properties which can predict safe approximations to program behaviors.

Early works [9–12] have taken the view that static analysis problems can be reduced to model checking. It is shown in [9, 10] that *data flow analysis* can be specified in a sublanguage of the modal μ-calculus [6] so that data flow equations can be implemented by evaluating a specific model checker. The results in [11, 12] show that data flow analysis can be reduced to model checking of a variant of Computation Tree Logic (CTL [1]).

In the other direction, recent research [13] presents a flow logic approach to static analysis which encodes model checking of *Action Computation Tree Logic* [14] formulas in *Alternation-Free Least Fixed Point Logic* (ALFP [15]). Similar work can be easily developed to reduce CTL model checking to static analysis. To be precise, it is shown that the set of states satisfying a modal formula can be characterised as the least element in a Moore family of acceptable sets of states for the ALFP formulas encoding the static analysis.

Continuing this line of work, we show that the *Alternation-Free fragment of the μ-calculus* can be characterised in a similar way. To do this, we propose in Section 2 an Alternation-Free Normal Form (AFNF), where negations are

only applied to closed subformulas; the expressive power of closed formulas in AFNF is equivalent to the alternation-free fragment of the μ-calculus. It is then shown in Section 3 that model checking for the alternation-free μ-calculus can be encoded in ALFP with the usual notion of *stratification*, i.e. the Moore family result makes use of a lexicographic ordering imposed by a suitable choice of ranking of the relations in the ALFP formula.

When negations are applied to open μ-calculus subformulas, our encoding method fails. We therefore establish a negative result in Section 4 showing that there exists a μ-calculus formula of alternation depth 2 whose least fixed point semantics cannot be characterized as a Moore Family property with respect to any notion of ranking. While static analysis can be developed in a fixed point setting (e.g. [16]) rather than in a Moore family setting this suggests that the majority of approaches to static analysis using abstract interpretation ideas are somehow more limited than model checking for logics that allow alternation.

2 The Modal μ-calculus

2.1 Kripke Structures

Kripke structures can be used to describe the behaviors of finite-state systems. The definition of *Kripke Structure* is modified slightly in comparison with [1] to distinguish different transitions in a system. Here, a Kripke structure over a set $\boldsymbol{P}$ of atomic propositions is a tuple $M = (S, T, L)$, where S is a set of states, T is a set of transition relations, and $L : S \to 2^{\boldsymbol{P}}$ labels each state with the set of true atomic propositions. Each element a in T is a transition relation and $a \subseteq S \times S$. As in [1] we also assume that the Kripke structure is total, although this is not necessary for our development.

2.2 Syntax and Semantics of the Modal μ-calculus

Definition 1 (Syntax of the Modal μ-calculus). *Let Var be a set of variables, and $\boldsymbol{P}$ be a set of atomic propositions. The syntax of modal μ-calculus formulas is defined as follows:*

$$\phi ::= \ p \mid Q \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \langle a \rangle \phi \mid [a]\phi \mid \mu Q.\phi \mid \nu Q.\phi$$

Here $p \in \mathbf{P}$, $Q \in Var$ and $a \in T$. The μ (resp. ν) operator is the least (resp. greatest) fixed point operator. For $\mu Q.\phi$ and $\nu Q.\phi$, it is required that all occurrences of Q in ϕ are under *an even number of negations* within ϕ. In this case, ϕ is said to be *syntactically monotone* in Q. If a variable is not bound by any fixed point operator in a formula, the variable is called a *free variable*. A formula is *closed* if there are no free variables in it.

A formula ϕ is interpreted as the set of states, on a given Kripke structure, that make it true and this set of states is denoted $[\![\phi]\!]_e$, where $e : Var \to 2^S$ is an environment. We use $e[Q \mapsto W]$ to denote the new environment updated from e by binding the relational variable Q to the set of states $W \subseteq S$. The semantics of μ-calculus formulas are defined as follows.

- $[\![p]\!]_e = \{\ s \mid p \in L(s)\ \}$
- $[\![Q]\!]_e = e(Q)$
- $[\![\neg\phi]\!]_e = S \setminus [\![\phi]\!]_e$
- $[\![\phi_1 \vee \phi_2]\!]_e = [\![\phi_1]\!]_e \cup [\![\phi_2]\!]_e$
- $[\![\phi_1 \wedge \phi_2]\!]_e = [\![\phi_1]\!]_e \cap [\![\phi_2]\!]_e$
- $[\![\langle a\rangle\phi]\!]_e = \{\ s \mid \exists s' :\ (s,\ s') \in a$ `and` $s' \in [\![\phi]\!]_e\}$
- $[\![[a]\phi]\!]_e = \{\ s \mid \forall s' :\ (s,\ s') \in a$ `implies` $s' \in [\![\phi]\!]_e\}$
- $[\![\mu Q.\phi]\!]_e$ is the least fixpoint of the function $\tau : 2^S \to 2^S$ defined by $\tau(W) = [\![\phi]\!]_{e[Q\mapsto W]}$
- $[\![\nu Q.\phi]\!]_e$ is the greatest fixpoint of the function $\tau : 2^S \to 2^S$ defined by $\tau(W) = [\![\phi]\!]_{e[Q\mapsto W]}$

The boolean operators have the usual meanings. If $(s,\ s') \in a$, we call s' an a-derivative of s. Due to the restricted use of negations in ϕ, monotonicity is guaranteed [1] for the function $\tau(W) = [\![\phi]\!]_{e[Q\mapsto W]}$.

A formula is in *Positive Normal Form* (PNF)[2] if all negations are only applied to atomic propositions and no variable is quantified twice. We give the syntax of the μ-calculus in Negation-Free PNF as follows.

Definition 2. *Let Var be a set of variables, $\boldsymbol{P}$ be a set of atomic propositions that is closed under negation. The syntax of the μ-calculus in Negation-Free PNF is defined as follows:*

$$\phi ::= \ p \mid Q \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \langle a\rangle\phi \mid [a]\phi \mid \mu Q.\phi \mid \nu Q.\phi$$

where no variable is quantified twice.

Lemma 1. *Every closed μ-calculus formula can be transformed to its Negation-Free PNF provided that the set $\boldsymbol{P}$ of atomic propositions is closed under negation.*

Model Checking for the μ-calculus is to find the set of states, on a given Kripke structure, that satisfy the μ-calculus formula ϕ according to the semantics ($[\![\phi]\!]_e$).

2.3 The Alternation Depth of the μ-calculus

Definitions of the alternation depth for modal μ-calculus formulas can be found in [2–4]. Based on [4], where the definition of the alternation depth is given for

a version of the modal μ-calculus with simultaneous fixpoints, we give our definition for the modal μ-calculus with just unary fixpoints.

We say that a formula φ is a *proper* subformula of formula ϕ iff φ is a subformula of ϕ but is not ϕ itself. A formula is called a μ-formula iff its main connective is μ. A subformula φ of ϕ is called a μ-subformula of it iff the main connective of φ is μ. The notions of ν-formula and ν-subformula can be defined similarly. Both μ-formula and ν-formula are called fixpoint formula, and similarly μ-subformula and ν-subformula are called fixpoint subformula. A μ-subformula φ of ϕ is called a *top-level* μ-subformula of it iff φ is not a μ-subformula of any other μ-subformula of ϕ. A μ-subformula φ of ϕ is called a *top* μ-subformula of it iff φ is not a μ-subformula of any other fixpoint subformula of ϕ. The notions of *top-level* ν-subformula and *top* ν-subformula can be defined similarly. Given a set of μ-calculus formulas, a formula in the set is called a *maximal* formula of the set iff it is not a proper subformula of any other formulas in this set.

Definition 3 (The Alternation Depth of Formulas). *For a closed μ-calculus formula ϕ given in Negation-Free PNF, the alternation depth, $ad(\phi)$, is defined inductively as follows (assuming that $max\{\emptyset\} = 0$).*

1. *If ϕ contains closed proper fixpoint subformulas, and ϕ_1,...,ϕ_n are the maximal formulas of the set of closed proper fixpoint subformulas of ϕ, then*

$$ad(\phi) = max(ad(\phi'), ad(\phi_1), ..., ad(\phi_n))$$

   where ϕ' is obtained from ϕ by substituting new atomic propositions p_1,...,p_n for ϕ_1,...,ϕ_n.
2. *If ϕ contains no closed proper fixpoint subformulas then $ad(\phi)$ is defined as follows.*
   - *$ad(p) = 0$, for any atomic proposition p.*
   - *$ad(\phi_1 \vee \phi_2) = ad(\phi_1 \wedge \phi_2) = max(ad(\phi_1),\ ad(\phi_2))$.*
   - *$ad([a]\varphi) = ad(\langle a\rangle\varphi) = ad(\varphi)$, for any transition relation a.*
   - *$ad(\mu Q.\varphi) = 1 + max\{ad(\varphi'_1), ..., ad(\varphi'_n)\}$ where $\varphi_i (1 \leq i \leq n)$ is top-level ν-subformula of φ and $\varphi'_i (1 \leq i \leq n)$ is constructed from φ_i by substituting all free variables with any new propositions.*
   - *$ad(\nu Q.\varphi) = 1 + max\{ad(\varphi'_1), ..., ad(\varphi'_n)\}$ where $\varphi_i (1 \leq i \leq n)$ is top-level μ-subformula of φ and $\varphi'_i (1 \leq i \leq n)$ is constructed from φ_i by substituting all free variables with any new propositions.*

As in [3], we define the *alternation-free* fragment of μ-calculus formulas as those formulas whose alternation depth are zero or one.

2.4 The Alternation-Free Fragment of the μ-Calculus

In this section, we propose an Alternation-Free Normal Form (AFNF) and show that closed formulas in AFNF exactly characterize the alternation-free fragment of the modal μ-calculus. This will facilitate our subsequent development.

Definition 4 (Syntax of Alternation-Free Normal Form). *Let Var be a set of variables, $\boldsymbol{P}$ be a set of atomic propositions that is closed under negation. The syntax of Alternation-Free Normal Form is defined as follows:*

$$\phi ::= \ p \mid Q \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \langle a \rangle \phi \mid [a]\phi \mid \mu Q.\phi \mid \neg \mu Q.\phi$$

where no variable is quantified twice and $\neg \mu Q.\phi$ is a closed formula.

We are most interested in closed formulas in AFNF and have the following lemmas.

Lemma 2. *Every alternation-free μ-calculus formula ϕ in Negation-Free PNF can be translated to its Alternation-Free Normal Form ϕ' while preserving the semantics.*

Lemma 3. *Every μ-calculus formula ϕ' in Negation-Free PNF translated from a closed formula ϕ in Alternation-Free Normal Form is alternation-free.*

The following proposition is the main result of this section.

Proposition 1. *Closed formulas defined in Alternation-Free Normal Form exactly characterize the alternation-free fragment of modal μ-calculus formulas.*

3 Alternation-Free μ-calculus in ALFP

3.1 Alternation-Free Least Fixed Point Logic

Alternation-Free Least Fixed Point Logic [15] has proved to be very useful for expressing static analyses in a general form that can easily be implemented. Given a fixed countable set X of variables and a finite alphabet $\mathcal{R}$ of predicate symbols, we define the syntax of ALFP as follows.

$$\begin{aligned} v \quad &::= c \mid x \\ pre &::= R(v_1, ..., v_n) \mid \neg R(v_1, ..., v_n) \mid pre_1 \wedge pre_2 \\ &\quad \mid pre_1 \vee pre_2 \mid \forall x : \ pre \mid \exists x : \ pre \\ cl \quad &::= R(v_1, ..., v_n) \mid \textbf{true} \mid cl_1 \wedge cl_2 \mid pre \Rightarrow cl \mid \forall x : \ cl \end{aligned}$$

The preconditions and clauses are interpreted over a finite and non-empty universe $\mathcal{U}$. The constant c is an element of $\mathcal{U}$, the variable $x \in X$ ranges over $\mathcal{U}$, and the n-ary relation $R \in \mathcal{R}$ denotes a subset of $\mathcal{U}^n$.

An *occurrence* of a relation R in a clause is a subformula of the form $R(v_1, ..., v_n)$. If it occurs in a precondition and is not negated, it is a *positive use*. If it occurs in a precondition and is negated, i.e. has the form $\neg R(v_1, ..., v_n)$, it is a *negative*

use. All other occurrences are *definitions* and often occur to the right of an implication. To ensure the existence of a least model, we shall pay special attention to the negative uses of relations. We restrict ourselves to the *stratified* fragment of clauses. The notion of stratification is given as follows.

A clause cl is *stratified* if there is a number r, an assignment of numbers called ranks $\texttt{rank}_R \in \{0, ..., r\}$ to each relation R, and a way to write the clause cl in the form $\bigwedge_{0 \leq i \leq r} cl_i$ such that the following holds for all clauses:

- if cl_i contains a definition of R then $\texttt{rank}_R = i$;
- if cl_i contains a positive use of R then $\texttt{rank}_R \leq i$; and
- if cl_i contains a negative use of R then $\texttt{rank}_R < i$.

Example 1. The following clause is not in ALFP since it is ruled out by the notion of stratification:

$$(\forall x : R_1(x) \Rightarrow R_2(x)) \wedge (\forall x : \neg R_2(x) \Rightarrow R_1(x))$$

This is because it is not possible that we have both $\texttt{rank}_{R_1} \leq \texttt{rank}_{R_2}$ and $\texttt{rank}_{R_2} < \texttt{rank}_{R_1}$.

The interpretation of ALFP is given in Table 1 in terms of satisfaction relations

$$(\varrho, \sigma)\ \underline{\texttt{sat}}\ pre \qquad \texttt{and} \qquad (\varrho, \sigma)\ \underline{\texttt{sat}}\ cl$$

where ϱ is the interpretation of relations and σ is the interpretation of variables. We write $\varrho(R)$ for the set of k-tuples $(a_1, ...a_k)$ from $\mathcal{U}$ associated with the k-ary predicate R, we use $\sigma(x)$ to denote the atom of $\mathcal{U}$ bound to x and $\sigma[x \mapsto a]$ stands for the mapping that is σ except that x is mapped to a. We also treat a constant c as a variable by setting $\sigma(c) = c$.

A clause with no free variables is called *closed*, and in closed clauses the interpretation σ is of no importance. For a fixed interpretation σ_0, when cl is closed, we have that $(\varrho, \sigma)\ \underline{\texttt{sat}}\ cl$ agrees with $(\varrho, \sigma_0)\ \underline{\texttt{sat}}\ cl$.

According to the choice of ranks we have made, we define a lexicographic ordering, $\sqsubseteq$, for the interpretations of relations, ϱ, as follows: $\varrho_1 \sqsubseteq \varrho_2$ if there exists a rank $i \in \{0, ..., r\}$ such that (1) $\varrho_1(R) = \varrho_2(R)$ whenever $rank(R) < i$, (2) $\varrho_1(R) \subseteq \varrho_2(R)$ whenever $rank(R) = i$, and (3) either $i = r$ or $\varrho_1(R) \subset \varrho_2(R)$ for some R with $rank(R) = i$. We define $\varrho_1 \subseteq \varrho_2$ to mean $\varrho_1(R) \subseteq \varrho_2(R)$ for all $R \in \mathcal{R}$.

The set of interpretations of relations constitutes a complete lattice with respect to $\sqsubseteq$. Moreover, we know from [15] that the set of solutions to an ALFP clause constitutes a Moore Family. Recall that a Moore Family [7] is a subset Y of a complete lattice $L = (L, \sqsubseteq)$ that is closed under greatest lower bounds: $\forall Y' \subseteq Y : \bigsqcap Y' \in Y$. The Moore Family result of ALFP is given as follows:

Table 1. Interpretation of ALFP

(ϱ,σ) $\underline{\texttt{sat}}$ $R(v_1,...,v_n)$	iff $(\sigma(v_1),...,\sigma(v_n)) \in \varrho(R)$
(ϱ,σ) $\underline{\texttt{sat}}$ $\neg R(v_1,...,v_n)$	iff $(\sigma(v_1),...,\sigma(v_n)) \notin \varrho(R)$
(ϱ,σ) $\underline{\texttt{sat}}$ $pre_1 \wedge pre_2$	iff (ϱ,σ) $\underline{\texttt{sat}}$ pre_1 and (ϱ,σ) $\underline{\texttt{sat}}$ pre_2
(ϱ,σ) $\underline{\texttt{sat}}$ $pre_1 \vee pre_2$	iff (ϱ,σ) $\underline{\texttt{sat}}$ pre_1 or (ϱ,σ) $\underline{\texttt{sat}}$ pre_2
(ϱ,σ) $\underline{\texttt{sat}}$ $\forall x:\ pre$	iff $(\varrho,\sigma[x \mapsto a])$ $\underline{\texttt{sat}}$ pre for all $a \in \mathcal{U}$
(ϱ,σ) $\underline{\texttt{sat}}$ $\exists x:\ pre$	iff $(\varrho,\sigma[x \mapsto a])$ $\underline{\texttt{sat}}$ pre for some $a \in \mathcal{U}$
(ϱ,σ) $\underline{\texttt{sat}}$ $R(v_1,...,v_n)$	iff $(\sigma(v_1),...,\sigma(v_n)) \in \varrho(R)$
(ϱ,σ) $\underline{\texttt{sat}}$ **true**	iff **true**
(ϱ,σ) $\underline{\texttt{sat}}$ $cl_1 \wedge cl_2$	iff (ϱ,σ) $\underline{\texttt{sat}}$ cl_1 and (ϱ,σ) $\underline{\texttt{sat}}$ cl_2
(ϱ,σ) $\underline{\texttt{sat}}$ $pre \Rightarrow cl$	iff (ϱ,σ) $\underline{\texttt{sat}}$ cl whenever (ϱ,σ) $\underline{\texttt{sat}}$ pre
(ϱ,σ) $\underline{\texttt{sat}}$ $\forall x:\ cl$	iff $(\varrho,\sigma[x \mapsto a])$ $\underline{\texttt{sat}}$ cl for all $a \in \mathcal{U}$

Proposition 2. *The set* $\{\varrho|(\varrho,\sigma_0)\ \underline{\texttt{sat}}\ cl\}$ *is a Moore Family, i.e. is closed under greatest lower bounds, whenever* cl *is closed and stratified; the greatest lower bound* $\sqcap\ \{\varrho|(\varrho,\sigma_0)\ \underline{\texttt{sat}}\ cl\}$ *is the least model of* cl*.*

More generally, given ϱ_0 *the set* $\{\varrho|(\varrho,\sigma_0)\ \underline{\texttt{sat}}\ cl \wedge \varrho_0 \subseteq \varrho\}$ *is a Moore Family and* $\sqcap\ \{\varrho|(\varrho,\sigma_0)\ \underline{\texttt{sat}}\ cl \wedge \varrho_0 \subseteq \varrho\}$ *is the least model.*

3.2 The Alternation-Free Fragment of the μ-Calculus in ALFP

We encode the model checking problem for the alternation-free μ-calculus into ALFP. According to Proposition 1, we use closed formulas defined in Alternation-Free Normal Form to characterize the alternation-free fragment of the μ-calculus.

We first encode a Kripke structure $M = (S, T, L)$ into ALFP by defining corresponding relations as follows. Assume that the universe is $\mathcal{U} = S$,

- for each atomic proposition p we define a predicate P_p such that $\varrho_0(P_p)(s)$ if and only if $p \in L(s)$, and
- for each element a in T, we define a binary relation a such that $\varrho_0(T_a)(s,t)$ if and only if $(s,t) \in a$.

We are most interested in variables in a μ-calculus formula. Therefore, we define only relations for all variables that occur in a given formula. We first introduce the idea of *Strongly Benign Translation* as follows.

Definition 5. *A Strongly Benign Translation is a translation from a* μ*-calculus formula* ϕ *to an ALFP clause* cl *such that we define a relation* R_Q *in* cl *iff* Q *is a variable in* ϕ*.*

To develop a Strongly Benign Translation for the alternation-free fragment of the μ-calculus, for each μ-calculus formula ϕ, we map it to a pair $\langle cl_\phi, pre_\phi \rangle$,

Table 2. Strongly Benign Translation of the Alternation-Free μ-calculus in ALFP

$$
\begin{array}{lcl}
p & \longmapsto & \langle \mathbf{true},\ P_p(s)\rangle \\
Q & \longmapsto & \langle \mathbf{true},\ R_Q(s)\rangle \\
\phi_1 \vee \phi_2 & \longmapsto & \langle cl_{\phi_1} \wedge cl_{\phi_2},\ pre_{\phi_1} \vee pre_{\phi_2}\rangle \\
 & & \texttt{whenever } \phi_1 \longmapsto \langle cl_{\phi_1},\ pre_{\phi_1}\rangle \texttt{ and } \phi_2 \longmapsto \langle cl_{\phi_2},\ pre_{\phi_2}\rangle \\
\phi_1 \wedge \phi_2 & \longmapsto & \langle cl_{\phi_1} \wedge cl_{\phi_2},\ pre_{\phi_1} \wedge pre_{\phi_2}\rangle \\
 & & \texttt{whenever } \phi_1 \longmapsto \langle cl_{\phi_1},\ pre_{\phi_1}\rangle \texttt{ and } \phi_2 \longmapsto \langle cl_{\phi_2},\ pre_{\phi_2}\rangle \\
\langle a\rangle\phi & \longmapsto & \langle cl_\phi,\ \exists s' : T_a(s,s') \wedge pre_\phi[s'/s]\rangle \\
 & & \texttt{whenever } \phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle \\
[a]\phi & \longmapsto & \langle cl_\phi,\ \forall s' : \neg T_a(s,s') \vee pre_\phi[s'/s]\rangle \\
 & & \texttt{whenever } \phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle \\
\mu Q.\phi & \longmapsto & \langle [\forall s : pre_\phi \Rightarrow R_Q(s)] \wedge cl_\phi,\ R_Q(s)\rangle \\
 & & \texttt{whenever } \phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle \\
\neg\mu Q.\phi & \longmapsto & \langle cl_{\mu Q.\phi},\ \neg R_Q(s)\rangle \\
 & & \texttt{whenever } \mu Q.\phi \longmapsto \langle cl_{\mu Q.\phi},\ pre_{\mu Q.\phi}\rangle
\end{array}
$$

where cl_ϕ is an ALFP clause and pre_ϕ is a precondition in ALFP. We use $pre_\phi[s'/s]$ to denote a precondition resulting from pre_ϕ by substituting the free variable s in pre_ϕ with s'. Assume ϱ is the least model of cl_ϕ subject to $\varrho(R_{Q_1}) \supseteq S_1, ..., \varrho(R_{Q_n}) \supseteq S_n, \varrho \supseteq \varrho_0$, where ϱ_0 defines P_p and T_a and $Q_1, ..., Q_n$ are all the free variables in ϕ. The intention of our development is that $s' \in [\![\phi]\!]_{e[Q_1 \mapsto S_1, ..., Q_n \mapsto S_n]}$ iff $(\varrho, \sigma[s \mapsto s'])$ $\underline{\texttt{sat}}$ pre_ϕ, and that when ϕ takes the form $\mu Q.\phi$, we have that $[\![\mu Q.\phi]\!]_{e[Q_1 \mapsto S_1, ..., Q_n \mapsto S_n]}$ equals $\varrho(R_Q)$. The Strongly Benign Translation we have developed is given in Table 2.

For atomic proposition p, we simply define cl_p as **true** since there are no bounded variables in p. We make use of the predefined predicate P_p and define pre_p as $P_p(s)$. For a variable Q, we also define cl_Q as **true** since the Q is a free variable here. We define pre_Q as $R_Q(s)$.

For $\phi_1 \vee \phi_2$, we assume that $\phi_1 \longmapsto \langle cl_{\phi_1},\ pre_{\phi_1}\rangle$ and $\phi_2 \longmapsto \langle cl_{\phi_2},\ pre_{\phi_2}\rangle$. This means that for each subformula $\mu Q.\phi$ in ϕ_1 (or ϕ_2), the relation R_Q is defined correctly in cl_{ϕ_1} (or cl_{ϕ_2}) and that pre_{ϕ_1} and pre_{ϕ_2} are also defined as expected. We define $cl_{\phi_1 \vee \phi_2}$ as $cl_{\phi_1} \wedge cl_{\phi_2}$. This ensures that for each subformula $\mu Q.\phi$ in $\phi_1 \vee \phi_2$, R_Q is defined correctly in $cl_{\phi_1} \wedge cl_{\phi_2}$. It's also natural to define $pre_{\phi_1 \vee \phi_2}$ as $pre_{\phi_1} \vee pre_{\phi_2}$. The case for $\phi_1 \wedge \phi_2$ follows the same pattern.

For $\langle a\rangle\phi$, we assume that $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$. This means that for each subformula $\mu Q.\varphi$ in ϕ, the relation R_Q is defined correctly in cl_ϕ and that pre_ϕ is also defined in an intended way. We simply define $cl_{\langle a\rangle\phi}$ to be the same as cl_ϕ since this suffices to guarantee that for each subformula $\mu Q.\varphi$ in $\langle a\rangle\phi$, the relation R_Q is defined correctly in $cl_{\langle a\rangle\phi}$. We define $pre_{\langle a\rangle\phi}$ as $\exists s' : T_a(s,s') \wedge pre_\phi[s'/s]$. This means for any state s if $pre_\phi[s'/s]$ holds on any of the a-derivative s' of s,

then $pre_{\langle a\rangle\phi}$ holds on state s. This matches the semantics for $\langle a\rangle\phi$.

For $[a]\phi$, we also assume that $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$. For a similar reason as in the case for $\langle a\rangle\phi$, we define $cl_{[a]\phi}$ to be the same as cl_ϕ. We define $pre_{[a]\phi}$ as $\forall s' : \neg T_a(s, s') \vee pre_\phi[s'/s]$. This means for any state s if $pre_\phi[s'/s]$ holds on all of the a-derivative s' of s, then $pre_{[a]\phi}$ holds on state s. Notice here that if s has no a-derivatives, $pre_{[a]\phi}$ still holds on s. This also matches the semantics for $[a]\phi$.

For $\mu Q.\phi$, we assume that $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$ as well. We define $cl_{\mu Q.\phi}$ as $[\forall s : pre_\phi \Rightarrow R_Q(s)] \wedge cl_\phi$. The first conjunct $[\forall s : pre_\phi \Rightarrow R_Q(s)]$ defines the relation R_Q and the second conjunct cl_ϕ ensures that for each subformula $\mu Q'.\varphi$ in ϕ, the relation $R_{Q'}$ is also defined correctly in $[\forall s : pre_\phi \Rightarrow R_Q(s)] \wedge cl_\phi$. The mapping here matches the semantics for the least fixed point operator μ. We define $pre_{\mu Q.\phi}$ as $R_Q(s)$.

For $\neg\mu Q.\phi$, we assume that $\mu Q.\phi \longmapsto \langle cl_{\mu Q.\phi},\ pre_{\mu Q.\phi}\rangle$. We define $cl_{\neg\mu Q.\phi}$ to be the same as $cl_{\mu Q.\phi}$. This guarantees that for each subformula $\mu Q'.\varphi$ in $\mu Q.\phi$, the relation $R_{Q'}$ is also defined correctly in $cl_{\neg\mu Q.\phi}$. We simple define $pre_{\neg\mu Q.\phi}$ as $\neg R_Q(s)$.

We have the following lemma.

Lemma 4. *Given a closed μ-calculus formula ϕ in AFNF, assume that $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$ holds according to Table 2, the clause cl_ϕ is closed and stratified.*

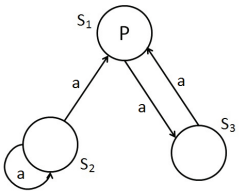
The following theorem shows that the precondition pre_ϕ in our mapping $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$ correctly characterizes the semantics of ϕ.

Theorem 1. *Let ϕ be a μ-calculus formula in Alternation-Free Normal Form with $Q_1, ..., Q_n$ being all the free variables in it. Assume that $\phi \longmapsto \langle cl_\phi,\ pre_\phi\rangle$. For the least solution ϱ of cl_ϕ such that $\varrho = \sqcap\{\varrho \mid (\varrho, \sigma)\ \underline{\mathtt{sat}}\ cl_\phi \wedge \varrho(R_{Q_1}) \supseteq S_1, ..., \wedge\ \varrho(R_{Q_n}) \supseteq S_n \wedge \varrho \supseteq \varrho_0\}$, where ϱ_0 defines P_p and T_a, we have $s' \in [\![\phi]\!]_{e[Q_1\mapsto S_1,...,Q_n\mapsto S_n]}$ iff $(\varrho, \sigma[s \mapsto s'])\ \underline{\mathtt{sat}}\ pre_\phi$.*

We focus on alternation-free μ-calculus formulas of the form $\mu Q.\phi$. This is not a restriction since $[\![\phi]\!] = [\![\mu Q.\phi]\!]$ when Q is not a free variable in ϕ. From Theorem 1, we have the following corollary saying that the best analysis result of our approach for the alternation-free μ-calculus coincides with the solution for the corresponding model checking problem.

Corollary 1. *Let $\mu Q.\phi$ be a closed μ-calculus formula in Alternation-Free Normal Form. Assume that $\mu Q.\phi \longmapsto \langle cl_{\mu Q.\phi},\ pre_{\mu Q.\phi}\rangle$. For the least model ϱ of $cl_{\mu Q.\phi}$ such that $\varrho = \sqcap\{\varrho | (\varrho, \sigma)\ \underline{\mathtt{sat}}\ cl_{\mu Q.\phi}, \varrho \supseteq \varrho_0\}$, where ϱ_0 defines P_p and T_a, we have $[\![\mu Q.\phi]\!] = \varrho(R_Q)$.*

Example 2. Consider a Kripke structure, given by the diagram to the left, where $S = \{s_1, s_2, s_3\}$, the transition relation $T = \{a\}$ is represented by edges labeled with a between states, and L labels s_1 with proposition p.



$\varrho(R_Q)$	$[\![\mu Q.[a](p \vee Q)]\!]$
$\{s_1, s_3\}$	$\{s_1, s_3\}$

We evaluate the formula $\mu Q.[a](p \vee Q)$ over the above Kripke structure using ALFP and the semantics of the μ-calculus respectively. The results are given in the table to the right.

In our static analysis approach, we will first encode the above Kripke structure in ϱ_0 and then generate the clause $cl_{\mu Q.[a](p \vee Q)}$ for the formula $\mu Q.[a](p \vee Q)$ according to Table 2. We list this process as follows. The least solution ϱ to $cl_{\mu Q.[a](p \vee Q)}$ subject to $\varrho_0 \subseteq \varrho$ can be calculated by succinct solver [15].

ϕ	cl_ϕ	pre_ϕ
p	**true**	$P_p(s)$
Q	**true**	$R_Q(s)$
$p \vee Q$	**true** $\wedge$ **true**	$P_p(s) \vee R_Q(s)$
$[a](p \vee Q)$	**true** $\wedge$ **true**	$\forall s' : \neg T_a(s, s') \vee P_p(s') \vee R_Q(s')$
$\mu Q.[a](p \vee Q)$	$\forall s : pre_{[a](p \vee Q)} \Rightarrow R_Q(s) \wedge$ **true** $\wedge$ **true**	$R_Q(s)$

4 Stratification Fails to Capture Syntactic Monotonicity

In this section, we analyze μ-calculus formulas of alternation depth 2 with the model checking approach and the approach we developed in Section 3.2 respectively. The main result of this section is that the solution to the model checking problem for μ-calculus formulas of alternation depth 2 cannot be characterised by a Moore Family result.

To encode a closed μ-calculus formula ϕ into ALFP, we shall assume there must exist a clause defining the relation R_Q for each variable Q in ϕ. We focus on the rank of R_Q. We explain our negative result as follows in a more general way where we assign a rank to each variable Q in ϕ.

Given a formula ϕ of the μ-calculus and let the list of subformulas $\overrightarrow{\phi}$ be some ordering of all fixpoint subformulas of ϕ, i.e. $\overrightarrow{\mu Q.\mu R.(Q \vee R)} = (\mu Q.\mu R.(Q \vee$

$R), \mu R.(Q \vee R))$. The model checking semantics of ϕ easily extends to $\overrightarrow{\phi}$, i.e. $[\![\overrightarrow{\mu Q.\mu R.(Q \vee R)}]\!] = ([\![\mu Q.\mu R.(Q \vee R)]\!], [\![\mu R.(Q \vee R)]\!]_{[Q \mapsto [\![\mu Q.\mu R.(Q \vee R)]\!]]})$.

Let ϕ be a closed formula of the μ-calculus. Assume that $\sigma Q_i.\phi_i$ (σ is either μ or ν) is a fixpoint subformula of ϕ ($1 \leq i \leq n$). We define the function $F : S^n \to S^n$ by $F(S_1, ..., S_n) = (\widetilde{[\![\phi_1]\!]_e}, ..., \widetilde{[\![\phi_n]\!]_e})$, where $e(Q_i) = S_i$, $\widetilde{\phi_i} = \phi_i[Q_j/\sigma Q_j.\phi_j]$ ($1 \leq j \leq n$), and $\sigma Q_j.\phi_j$ is a top fixpoint subformula of ϕ_i. The notation $\phi_i[Q_j/\sigma Q_j.\phi_j]$ refers to a formula resulting from ϕ_i by substituting $\sigma Q_j.\phi_j$ with Q_j. We have the following theorem.

Theorem 2. *There exists a μ-calculus formula ϕ of alternation depth 2, where $Q_1, ..., Q_n$ is some ordering of all the variables in ϕ, such that $[\![\overrightarrow{\phi}]\!] = (S_1, ..., S_n)$ is not the least solution to the equation $F(S_1, ..., S_n) = (S_1, ..., S_n)$ with respect to $\sqsubseteq$ for any choice of ranking.*

Proof. Let $M = (S, T, L)$ be a Kripke structure , where $S = \{s_1, s_2\}$, $T = \{a\}$, $a = \{(s_1, s_2), (s_2, s_2)\}$, and L labels s_2 with proposition p. Consider the formula $\phi = \mu Q.(\neg \mu R.(R \vee (\neg Q \wedge p)))$. We can see that $ad(\phi) = 2$ once we translate ϕ to its Negation-Free PNF.

We define $F(S_1, S_2) = ([\![\neg R]\!]_e, [\![R \vee (\neg Q \wedge p)]\!]_e)$, where $e(Q) = S_1$ and $e(R) = S_2$. Let's consider solutions to the equation $F(S_1, S_2) = (S_1, S_2)$. In the following, we use $\varrho(i)$ to denote the ith ($i = 1, 2$) component in ϱ.

Let $\overrightarrow{\phi} = (\mu Q.(\neg \mu R.(R \vee (\neg Q \wedge p))), \mu R.(R \vee (\neg Q \wedge p)))$. According to the model checking semantics, we know that $\varrho_0 = [\![\overrightarrow{\phi}]\!] = ([\![\mu Q.(\neg \mu R.(R \vee (\neg Q \wedge p)))]\!], [\![\mu R.(R \vee (\neg Q \wedge p))]\!]_{e[Q \mapsto [\![\mu Q.(\neg \mu R.(R \vee (\neg Q \wedge p)))]\!]]}) = (\{s_1\}, \{s_2\})$. It's obvious that ϱ_0 is a solution to the equation$F(S_1, S_2) = (S_1, S_2)$. We also have another two solutions $\varrho_1 = (\emptyset, \{s_1, s_2\})$ and $\varrho_2 = (\{s_1, s_2\}, \emptyset)$ to it as well.

Since both $\varrho_1(1) \subset \varrho_0(1)$ and $\varrho_2(2) \subset \varrho_0(2)$ hold, it's obvious that ϱ_0 is not the least solution to the equation $F(S_1, S_2) = (S_1, S_2)$ with respect to $\sqsubseteq$ for any choice of ranking.

Theorem 2 can be extended to the case of a μ-calculus formula ϕ of alternation depth n ($n > 2$). Whenever we develop a strongly benign translation to encode μ-calculus formulas to ALFP clauses, we implicitly define a function F above. Therefore, encoding the full μ-calculus formulas into ALFP using strongly benign translation is not feasible.

5 Conclusion

Based on our previous work [13], we have shown that model checking for the alternation-free μ-calculus can also be described as static analysis of modal logic.

Efficient translation of CTL into the μ-calculus can be found in [1, 2] and [2] points out that CTL can be encoded in Lμ_1, which is exactly the alternation-free fragment of the μ-calculus. Therefore, our approach can be used to deal with CTL as well, thereby generalizing also [13].

Our negative result is that the full μ-calculus cannot be encoded in a similar way regardless of the choice of ranking. Results in [9, 11] on the one hand, and our work on the other hand, have improved our understanding of the interplay between model checking and static analysis. It would be interesting to identify fragments of the modal μ-calculus that reside properly between alternation depth 2 and alternation free for which the ALFP-based development might still work, i.e. for which the least fixed point can be described as a Moore family result.

References

1. Edmund M. Clarke, Orna Grumberg, and Doron A.Peled. *Model Checking*. MIT Press, 1999.
2. E. Allen Emerson, Chin-Laung Lei: Efficient Model Checking in Fragments of the Propositional Mu-Calculus (Extended Abstract) *LICS 1986*: 267-278
3. Rance Cleaveland, Bernhard Steffen: A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus. *Formal Methods in System Design 2*(2): 121-147 (1993)
4. Henrik Reif Andersen: Model Checking and Boolean Graphs. *Theor. Comput. Sci. 126*(1): 3-30 (1994)
5. Christel Baier, Joost-Pieter Katoen: Principles of model checking. MIT Press 2008: I-XVII, 1-975
6. Dexter Kozen: Results on the Propositional mu-Calculus. *Theor. Comput. Sci. 27*: 333-354 (1983)
7. Flemming Nielson, Hanne Riis Nielson, Chris Hankin: Principles of program analysis (2. corr. print). Springer 2005: I-XXI, 1-452
8. Hanne Riis Nielson, Flemming Nielson: Flow Logic: A Multi-paradigmatic Approach to Static Analysis. *The Essence of Computation 2002*: 223-244
9. Bernhard Steffen: Data Flow Analysis as Model Checking. *TACS 1991*: 346-365
10. Bernhard Steffen: Generating Data Flow Analysis Algorithms from Modal Specifications. *Sci. Comput. Program. 21*(2): 115-139 (1993)
11. David A. Schmidt, Bernhard Steffen: Program Analysis as Model Checking of Abstract Interpretations. *SAS 1998*: 351-380
12. David A. Schmidt: Data Flow Analysis is Model Checking of Abstract Interpretations. *POPL 1998*: 38-48
13. Flemming Nielson, Hanne Riis Nielson: Model Checking Is Static Analysis of Modal Logic. *FOSSACS 2010*: 191-205
14. Rocco De Nicola, Frits W. Vaandrager: Action versus State based Logics for Transition Systems. *Semantics of Systems of Concurrent Processes 1990*: 407-419
15. Flemming Nielson, Helmut Seidl, Hanne Riis Nielson: A Succinct Solver for ALFP. *Nord. J. Comput. 9*(4): 335-372 (2002)
16. Flemming Nielson: Two-Level Semantics and Abstract Interpretation. *Theor. Comput. Sci. 69*(2): 117-242 (1989)